Rollback Plan

Rollback Plan

> **Project:** {{PROJECT_NAME}} **Version:** {{VERSION}} **Date:** {{DATE}}
> **Author:** {{AUTHOR}} **Status:** Draft | In Review | Approved **Reviewers:** {{REVIEWERS}}

Document History

Version	Date	Author	Changes
0.1	{{DATE}}	{{AUTHOR}}	Initial draft

Rollback Summary

Field	Value
Deployment being rolled back	v{{VERSION}}
Rollback target version	v{{ROLLBACK_VERSION}}
Rollback image / artifact	`{{ROLLBACK_IMAGE}}`
DB migration reversible	{{DB_REVERSIBLE}}
Estimated rollback time	{{ROLLBACK_TIME}} minutes
Rollback owner	{{ROLLBACK_OWNER}}
Backup to restore (if needed)	{{BACKUP_ID}} (taken at {{BACKUP_TIME}})

---

1. Rollback Decision Criteria

Roll back immediately if ANY of these conditions occur:

Trigger	Threshold	Measurement	Wait Before Deciding
Error rate spike	> {{ERROR_THRESHOLD}}%	Rolling 5-min average	{{WAIT_DURATION}} minutes
P99 latency spike	> {{P99_THRESHOLD}}ms sustained	Rolling 5-min P99	{{WAIT_DURATION}} minutes
Health check failures	> {{HEALTH_FAIL_PCT}}% instances	Load balancer health	0 minutes (immediate)
Smoke test failure	Any critical test fails	Automated smoke tests	0 minutes (immediate)
Data integrity issue	Any confirmed data corruption	Post-deploy verification	0 minutes (immediate)
Security vulnerability	Critical severity confirmed	Security alert	0 minutes (immediate)

Do NOT roll back for:

- Warning-level alerts that were present pre-deployment
- Increased error rate in non-critical paths < {{MINOR_ERROR_THRESHOLD}}%
- Expected behavior changes (verify against release notes first)
- Cosmetic/visual issues that don't affect functionality

---

2. Rollback Authority

Situation	Authority
Standard rollback (automated trigger)	On-call engineer (no approval needed)
Manual rollback (judgment call)	Senior engineer on duty
Business-hours manual rollback	Engineering Manager approval recommended
Off-hours manual rollback	On-call lead (inform manager post-rollback)

Authorization contact: {{ROLLBACK_AUTHORITY}} | {{PHONE}} | Slack: {{SLACK}}

---

3. Pre-Rollback Assessment

Data Changes Since Deployment

- **Deployment time:** {{DEPLOYMENT_TIME}}
- **Data changes since deployment:** {{DATA_CHANGES}}
- **Critical data at risk:** {{DATA_RISK}}

- **Acceptable to lose this data?** Yes / No / Needs analysis

Decision: Proceed with rollback / Rollback with data preservation steps / Do NOT rollback (data loss unacceptable)

Database Migration Reversibility

Migration	Type	Reversible	Down Migration Available
`{{MIGRATION_1}}`	{{TYPE}}	{{REVERSIBLE}}	{{AVAILABLE}}
`{{MIGRATION_2}}`	{{TYPE}}	{{REVERSIBLE}}	{{AVAILABLE}}

If migration is NOT reversible: Rollback requires database restore from backup (see Section 4.2)

External System State

System	Events Processed Since Deploy	Reversible	Action if Rollback
Payment gateway	{{PAYMENT_COUNT}} transactions	No	No action — transactions stand
Email service	{{EMAIL_COUNT}} emails sent	No	No action — emails sent stand
Webhooks	{{WEBHOOK_COUNT}} delivered	No	Notify downstream systems

---

4. Rollback Procedures

4.1 Application Rollback (Step by Step)

Total estimated time: {{APP_ROLLBACK_TIME}} minutes

```
# Step 1: Announce rollback (required)
# Post in war room: "ROLLBACK initiated — v{{VERSION}} → v{{ROLLBACK_VERSION}}"

# Step 2: Trigger rollback deployment
# Option A — CI pipeline rollback:
{{CI_ROLLBACK_CMD}}
```

```
# Option B — Direct deployment with previous image:
{{DIRECT_ROLLBACK_CMD}}

# Step 3: Monitor rollback progress
{{MONITOR_CMD}}

# Step 4: Confirm rollback complete
curl {{URL}}/api/version  # Should return {{ROLLBACK_VERSION}}
```

Verification commands:

```
# Check all instances running rollback version
{{INSTANCE_CHECK_CMD}}

# Check health
curl {{URL}}/health

# Check error rate (should drop immediately)
{{ERROR_RATE_CMD}}
```

4.2 Database Rollback (Migration Down)

Warning: Execute database rollback ONLY after confirming:

1. Application rollback is complete
2. Data loss from migration reversal is acceptable (see Section 3)
3. Down migration is available and tested

```
# Step 1: Confirm current migration state
{{MIGRATION_STATUS_CMD}}

# Step 2: Take emergency backup BEFORE running down migration
{{DB_BACKUP_CMD}}

# Step 3: Run down migration
{{DOWN_MIGRATION_CMD}}

# Step 4: Verify migration state
{{MIGRATION_VERIFY_CMD}}
```

```
# Step 5: Verify data integrity
bash scripts/verify-integrity.sh
```

If down migration fails or is not available: Restore from pre-deployment backup

```
# Restore from backup {{BACKUP_ID}}
{{DB_RESTORE_CMD}} --backup-id {{BACKUP_ID}}
```

4.3 Configuration Rollback

```
# Revert environment variables (if changed in this deployment)
{{CONFIG_ROLLBACK_CMD}}

# Verify configuration
{{CONFIG_VERIFY_CMD}}
```

Changed configuration to revert:

Variable	New Value (to revert FROM)	Previous Value (to revert TO)
`{{VAR_1}}`	`{{NEW_VALUE}}`	`{{OLD_VALUE}}`

4.4 DNS / CDN Rollback

DNS rollback (if DNS changes were made):

```
# Revert DNS record
{{DNS_REVERT_CMD}}

# Wait for propagation (TTL: {{DNS_TTL}}s)
sleep {{DNS_TTL}}

# Verify
nslookup {{DOMAIN}}
```

CDN cache purge (to clear cached version of new code):

```
{{CDN_PURGE_CMD}}
```

---

5. Verification After Rollback

Health Check Verification

- [ ] `GET {{URL}}/health` returns HTTP 200 with `{"status":"ok"}`
- [ ] `GET {{URL}}/health/ready` returns HTTP 200 (DB + Cache connected)
- [ ] All instances showing previous version: `{{VERSION_VERIFY_CMD}}`
- [ ] Load balancer health checks green for all instances

Smoke Test Execution

```
bash scripts/smoke-tests.sh {{ENVIRONMENT}}
```

- [ ] All critical smoke tests passing
- [ ] Critical user journey manually verified

Data Integrity Verification

```
bash scripts/verify-integrity.sh {{ENVIRONMENT}}
```

- [ ] No data loss confirmed (or data loss quantified and documented)
- [ ] Database in consistent state
- [ ] Replication lag normal

Monitoring Verification

- [ ] Error rate returned to pre-deployment baseline (< {{ERROR_BASELINE}}%)
- [ ] P99 latency returned to pre-deployment baseline
- [ ] No unexpected log errors
- [ ] Alerts silenced (if any were firing during incident)

---

6. Communication Plan

Internal Notification

Audience	Channel	When	Message
Engineering team	War room + Slack	At rollback initiation	"Rollback of v{{VERSION}} initiated"
Engineering management	Direct	At rollback decision	Summary of decision + expected timeline
Customer support	Slack	If user-facing impact	Support briefing note

External Notification

Audience	Channel	When	Trigger
Status page	{{STATUS_PAGE}}	At rollback initiation	Always (any production rollback)
Affected users	Email	If impact > {{EMAIL_THRESHOLD}}h	At rollback + recovery
SLA customers	Direct contact	Per contract	If SLA breach triggered

Status page message template:

```
We are currently experiencing an issue with {{PROJECT_NAME}} and have initiated a rollback
to resolve it. We expect service to be restored within {{EXPECTED_TIME}} minutes.
We apologize for the inconvenience and will provide updates every 15 minutes.
```

---

7. Post-Rollback Analysis

Post-rollback review scheduled: {{REVIEW_DATE}} **Post-mortem scheduled:** {{PM_DATE}} (within {{PM_SLA}}h of resolution)

Analysis questions:

1. What caused the rollback? (specific code/config/migration)
2. Could this have been detected earlier? (pre-production test coverage gap?)
3. Was the rollback executed correctly and quickly?
4. What process change would prevent this next time?

Output: Post-mortem document at post-mortem.md

---

8. Forward Fix vs Rollback Decision Matrix

Factor	Favors Forward Fix	Favors Rollback
Time to fix	< 30 min	> 30 min
DB migration	Not included	Included (rollback simpler)
Data written since deploy	Significant	Minimal
User impact severity	P3/P4	P1/P2
Fix risk	Low	High
Team availability	Senior dev available	Dev unavailable
Off-hours	Usually no	Usually yes

Default guideline: When uncertain, **rollback**. A rollback to a known good state is safer than a rushed forward fix.

Related Documents

- Deployment Checklist
- Release Notes
- Go-Live Runbook
- Post-Mortem
- Incident Report

Approval

Role	Name	Date	Signature
Author			
Reviewer			
Approver			

Revision #4
Created 2026-02-24 14:54:24 UTC by John
Updated 2026-06-28 20:03:50 UTC by John